UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/722,146	11/25/2003	Frank S. Caccavale	10830.0106NP	7027

27927 7590 09/24/2008
RICHARD AUCHTERLONIE
NOVAK DRUCE & QUIGG, LLP
1000 LOUISIANA
53RD FLOOR
HOUSTON, TX 77002

EXAMINER	
TRUONG, CAMQUY	
ART UNIT	**PAPER NUMBER**
2195	
MAIL DATE	**DELIVERY MODE**
09/24/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/722,146	CACCAVALE, FRANK S.	
	Examiner	**Art Unit**	
	CAMQUY TRUONG	2195	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE _3_ MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on _25 November 2003_.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) _1-36_ is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) _1-36_ is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _12/25/03_.

4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application

6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-36 are presented for examination.

Drawings

2. The drawings are objected to as failing to comply with 37 CFR 1.84(p)(5) because they include the following reference character(s) not mentioned in the description: reference number 44 of Figure 2 is not mentioned in the specification. Corrected drawing sheets in compliance with 37 CFR 1.121(d), or amendment to the specification to add the reference character(s) in the description in compliance with 37 CFR 1.121(b) are required in reply to the Office action to avoid abandonment of the application. Any amended replacement drawing sheet should include all of the figures appearing on the immediate prior version of the sheet, even if only one figure is being amended. Each drawing sheet submitted after the filing date of an application must be labeled in the top margin as either “Replacement Sheet” or “New Sheet” pursuant to 37 CFR 1.121(d). If the changes are not accepted by the examiner, the applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

Claim Rejections - 35 USC § 112

3. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter, which the applicant regards as his invention.

4. Claims 10 and 28, are rejected under 35 U.S.C 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

A. The following terms lack antecedence basis:

Claims 10 and 28, lines 1-4, recite the limitation “the distribution list” and “the randomized distribution list”.

B. The following claim language is indefinite:

Claims 10 and 28, lines 1-2, it recites “re-randomizing”. However, there is no “randomizing” step performed previously.

Claim Rejections - 35 USC § 103

. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. **Claims 1-6 and 19- 24 are rejected under 35 U.S.C. 103(a) as being unpatentable over Van Rietschote et al. (U.S. Patent 7,203,944 B1) in view of**

Greuel et al. (U.S. 7,003,564 B2) (herein after Greuel), and further in view of Baratz et al. (U.S. 2002/0034190 A1) (herein after Baratz).

6. As to claims 1 and 19, Van teaches the invention substantially as claimed including: in a data processing network including distributed processing units, a method comprising:

obtaining a respective utilization value of each distributed processing unit (the load (weight) of each virtual machine may be calculated according to the resource that it uses, col. 8, lines 35-37; and col. 10, lines 47-54). In order to calculate the load (weight) the resource usage of the virtual machine is obtained. Thus, Van inherently discloses a respective utilization value of each distributed processing unit is obtaining;

applying a function (formula 9, Fig. 4; col. 10, lines 47-49) to the respective utilization value of said each distributed processing unit to obtain a respective weight (load / weight, col. 11, lines 3-14) for said each distributed processing unit (the load on computer system 10A-10N may be the sum of the loads of the virtual machines executing on that computer system, col. 5, lines14-16, and the load of each virtual machine may be calculated according to the resource that it uses, col.8, lines 35-37/ the amount of CPU time used by the virtual machine (CPUTime in FIG. 4), the amount of I/O activity (I/O Activity in FIG. 4) generated by the virtual machine, and the amount of memory consumed by the virtual machine (Memory Pages in FIG. 4) are included in the load calculation, col. 10, lines 47-54); and

using the respective weights for the distributed processing units for distributing work requests to the distributed processing units so that the respective weight for said each distributed processing unit (if the requesting computer system's load exceeds the selected computer system's load, the virtual machine migration code may migrates the selected virtual machine to the selected computer system, col. 9, lines 3-9; col. 2, lines 16-24).

7. Van does not explicitly teach obtaining a respective weight by using a mapping function. However, Greuel teaches obtaining a respective weight by using a mapping function (for each system variable, a mapping by which a raw data value associated with the corresponding system variable is mapped to a score, and for each system variable, a weight, col. 2, lines 39-42 / a score mapped from the CPU utilization, col. 5, lines 4-29).

8. It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teaching of Van by incorporating the teaching of a respective weight by using a mapping function as taught by Greuel in order to gain the advantage of monitoring of the health or status of a computer network to determine whether the computer network is online or offline (i.e., up or down) as a result of providing reliable, available and high performing computer networks and applications.

9. Van and Greuel do not explicitly teach the respective weight for said each distributed processing unit specifies a respective frequency at which the work requests are distributed to said each distributed processing unit. However, Baratz teaches the respective weight for said each distributed processing unit specifies a respective frequency at which the work requests are distributed to said each distributed processing unit (monitoring the usage information from one or more of the network elements, based on this monitoring, the availability server predicts a forth coming time interval during which sufficient network elements will be available send a message to one or more user, paragraph 47-48).

10. It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teaching of Van and Greuel by incorporating the teaching of the respective weight for said each distributed processing unit specifies a respective frequency at which the work requests are distributed to said each distributed processing unit as taught by Baratz because this allow to monitor the available of idle resources in a network and use these resource for additional services.

11. As to claims 2 and 20, Greuel teaches wherein the respective utilization value of said each distributed processing unit is a percentage of saturation of said each distributed processing unit (Fig. 2B, CPU utilization present a percentage).

12. As to claims 3-4 and 21-22, Greuel teaches said each distributed processing

unit collects statistics for calculation of the respective utilization value of said each distributed processing unit (the data collector collects a raw data value corresponding to the system variable, col. 2, lines 22-23; and col. 3, lines 38-40).

13. As to claims 5 and 23, Greuel teaches wherein the respective weight for said each distributed processing unit is programmed into a mapping table (table 233, Fig. 2B; col. 5, lines 4-29), and the mapping function is applied to the respective utilization value of said each distributed processing unit to obtain the respective weight for said each distributed processing unit by indexing the mapping table with the respective utilization value of said each distributed processing unit to obtain the respective weight for said each distributed processing unit (the table 233 contains columns for the router name (or the address) (index), and the overall score in this example is computed as the weighted average of two numbers: (1) the interface health and (2) and a score mapped from the CPU utilization, col. 5, lines 4-10).

14. As to claims 6 and 24, they are rejected for the same reason as claim 1. In addition, Van teaches the function is selected to provide weights estimated to cause a balancing of loading upon the distributed processing units (randomly selects another computer system 10A 10N with which to compare loads. If the other computer system 10A -10N has a lower load, the VM migration code may migrate a virtual machine to the other computer system. The VM migration code on each computer system 10A 10N may be activated periodically, and may randomly select another computer system with

which to compare loads (weights) and to potentially migrate a virtual machine. Over time, the periodic random selecting by each computer system may lead to relative balance in the loads on the computer systems, col. 5, lines 22-37; and col. 8, lines 40-54).

15. **Claims 7-8 and 25-26 are rejected under 35 U.S.C. 103(a) as being unpatentable over Van Rietschote et al. (U.S. 7,203,944 B1) (hereinafter Van) in view of Greuel et al. (U.S. 7,003,564 B2) (herein after Greuel) and further in view of Baratz et al. (U.S. 2002/0034190 A1) (herein after Baratz) as applied to claims 1and 19 above, and further in view of Garnett et al. (U.S. 7,032,037) (herein after Garnett).**

16. As to claims 7 and 25, Van, Greuel and Baratz do not explicitly teach wherein the respective weights are used for weighted round-robin load balancing of the work requests upon the distributed processing units. However, Garnett teaches wherein the respective weights are used for weighted round-robin load balancing of the work requests upon the distributed processing units (load balancing algorithm is the "weighted round robin" where the number of connections assigned per server is specified by a weight assigned to each server, col. 32, lines 13-16).

17. It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teaching of Van, Greuel, and Baratz by incorporating

the teaching of the respective weights are used for weighted round-robin load balancing of the work requests upon the distributed processing units as taught by Garnett in order to gain the advantage of preventing a particular application on the server is overloaded as sugguest by Garnett.

18. As to claims 8 and 26, Garnett teaches the weighted round-robin load balancing performs round-robin load balancing when the weights are equal (all the servers have the same weight, col. 32, lines 16-18).

19. **Claims 9 and 27 are rejected under 35 U.S.C. 103(a) as being unpatentable over Van Rietschote et al. (U.S. 7,203,944 B1) (hereinafter Van) in view of Greuel et al. (U.S. 7,003,564 B2) (herein after Greuel) and further in view of Baratz et al. (U.S. 2002/0034190 A1) (herein after Baratz) as applied to claims 1 and 19 above, and further in view of Kapoor (U.S. 5,884,038).**

20. As to claims 9 and 27, Van, Greuel, and Baratz do not explicitly teach the respective weights for the distributed processing units are used for distributing work requests to the distributed processing units by creating a distribution list containing entries indicating the distributed processing units, the respective weight for said each distributed processing unit specifying the number of the entries indicating said each distributed processing unit, and by randomizing the distribution list, and accessing the

randomized distribution list for distributing the work requests to the distributed processing units as indicated by the entries in the randomized distribution list.

21. However, Kapoor teaches teach the respective weights for the distributed processing units are used for distributing work requests to the distributed processing units by creating a distribution list containing entries indicating the distributed processing units (an array A of numbers with W elements is created, the first w1 elements of array A are assigned a number, which corresponds to the web server with weight w1..., col. 5, lines 48-60 / the domain name server connects to each web server and removes the down web servers from the list, col. 5, lines 42-44 / after array A has been created, col. 5, lines 60-61) the respective weight for said each distributed processing unit specifying the number of the entries indicating said each distributed processing unit (the domain name server returns the IP address of a web server such that the total number of times that the IP address of each one of the web servers is returned in proportional to the relative weight of each web server, col. 5, lines 11-16), and by randomizing the distribution list (the domain name server randomize the list of web server, col. 5, lines 15-16 / the order of all the elements and array A are randomized, col. 5, lines 60-63), and accessing the randomized distribution list for distributing the work requests to the distributed processing units as indicated by the entries in the randomized distribution list (The domain name server returns the IP address of a web server such that the total number of times that the IP address of each one of the web servers is returned in proportional to the relative weight of each web

server, col. 2, lines 55-63). In order to return the IP address of a web server to servet client, the domain name server has to access to web server lists. Thus, it would have been obvious that the Kapoor teaches accessing the randomized distribution list for distributing the work requests to the distributed processing units as indicated by the entries in the randomized distribution list).

22. It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teaching of Van, Greuel, and Baratz by incorporating the teaching of creating a distribution list containing entries indicating the distributed processing units, the respective weight for said each distributed processing unit specifying the number of the entries indicating said each distributed processing unit, and by randomizing the distribution list, and accessing the randomized distribution list for distributing the work requests to the distributed processing units as indicated by the entries in the randomized distribution list as taught by Kapoor because this would allow to efficiently utilize the multiple web servers of an Internet host as well as reduce the skewed locking problems such that overall Internet traffic and response times are reduced (col. 2, lines 47-52).

23. **Claims 10 and 28 are rejected under 35 U.S.C. 103(a) as being unpatentable over Van Rietschote et al. (U.S. Patent 7,203,944 B1) in view of Greuel et al. (U.S. 7,003,564 B2) (herein after Greuel), further in view of Baratz et al. (U.S.**

2002/0034190 A1) (herein after Baratz) as applied to claims 1 and 19 above, and further in view of Grochowski (U.S. 6,115,807).

24. As to claims 10 and 28, Van, Greuel, Baratz do not explicitly teach re-randomizing the distribution list for re-use once the end of the distribution list is reached during the distributing of the work requests to the virus checking servers as indicated by the entries in the randomized distribution list. However, Grochowski teaches re-randomizing the distribution list for re-use once the end of the distribution list is reached during the distributing of the work requests to the virus checking servers as indicated by the entries in the randomized distribution list (if the decoder comes to the end of the queue, it rotates around to the beginning of queue (col.2, lines 65-67). As to the limitation of "randomize”, Grochowski teaches “rotating”. Noting that randomizer is also defined as rotator. Extrinsic evidence, inventor name (U.S. 6,819,276), disclose randomizer is also defined as rotator (col. 5, lines 46-47).

25. It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teaching of Van, Greuel, Garnett and kapoor by incorporating the teaching of re-randomizing the distribution list for re-use once the end of the distribution list is reached during the distributing of the work requests to the virus checking servers as indicated by the entries in the randomized distribution list as taught by Grochowski in order to gain advantage of determining how many of the decoded instructions may issue in a next clock cycle thereby to ensure that neither too many nor

too few instructions enter pipeline at any given time thereby improve the effective of processors.

26. **Claims 11 and 29 are rejected under 35 U.S.C. 103(a) as being unpatentable Van Rietschote et al. (U.S. 7,203,944 B1) (hereinafter Van) in view of Greuel et al. (U.S. 7,003,564 B2) (herein after Greuel), and further in view of Kapoor (U.S. 5,884,038).**

27. As to claims 11 and 29, Van teaches the invention substantially as claimed including: in a data processing network including distributed processing units, a method comprising:

obtaining a respective utilization value of each distributed processing unit (the load (weight) of each virtual machine may be calculated according to the resource that it uses, col. 8, lines 35-37; and col. 10, lines 47-54), in order to calculate the load (weight) the resource usage of the virtual machine is obtained, thus Van inherently discloses obtaining a respective utilization value of each distributed processing unit;

applying a function (formula 9, Fig. 4; and col. 10, lines 47-49) to the respective utilization value of said each distributed processing unit to obtain a respective weight (load / weight, col. 11, lines 3-14) for said each distributed processing unit (the load on computer system 10A-10N may be the sum of the loads of the virtual machines executing on that computer system, col. 5, lines14-16, and the load of each virtual machine may be calculated according to the resource that it uses, col.8, lines 35-37/ the

amount of CPU time used by the virtual machine (CPUTime in FIG. 4), the amount of I/O activity (I/O Activity in FIG. 4) generated by the virtual machine, and the amount of memory consumed by the virtual machine (Memory Pages in FIG. 4) are included in the load calculation, col. 10, lines 47-54); and

using the respective weights for the distributed processing units for distributing work requests to the distributed processing units so that the respective weight for said each distributed processing unit (if the requesting computer system’s load exceeds the selected computer system's load, the virtual machine migration code may migrates the selected virtual machine to the selected computer system, col. 9, lines 3-9; col. 2, lines 16-24).

28. Van does not explicitly teach obtaining a respective weight by using a mapping function. However, Greuel teaches obtaining a respective weight by using a mapping function (for each system variable, a mapping by which a raw data value associated with the corresponding system variable is mapped to a score, and for each system variable, a weight, col. 2, 39-42 / a score mapped from the CPU utilization, col. 5, lines 4-29).

29. It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teaching of Van by incorporating the teaching of a respective weight by using a mapping function as taught by Greuel in order to gain the advantage of monitoring of the health or status of a computer network to determine

whether the computer network is online or offline (i.e., up or down) as a result of providing reliable, available and high performing computer networks and applications.

30. Van and Greuel do not explicitly teach using the respective weights for the distributed processing units for producing a distribution list for distributing work requests to the distributed processing units for load balancing of the work requests upon the processing units, and repetitively randomizing the distribution list during the distribution of the work requests to the distributed processing units. However, Kapoor teaches using the respective weights for the distributed processing units for producing a distribution list (an array A of numbers with W elements is created, the first w1 elements of array A are assigned a number, which corresponds to the web server with weight w1..., col. 5, lines 48-60 / after array A has been created, col. 5, lines 60-61) for distributing work requests to the distributed processing units for load balancing of the work requests upon the processing units (provide IP addresses to client requesting an IP address base on the relative weights of each particular web server such that the workload is balanced between each of the web server, col. 5, lines 1-5) , and repetitively randomizing the distribution list during the distribution of the work requests to the distributed processing (the domain name server randomizes the list of web server, col. 5, lines 15-16).

31. It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teaching of Van and Greuel by incorporating the

teaching of using the respective weights for the distributed processing units for producing a distribution list for distributing work requests to the distributed processing units for load balancing of the work requests upon the processing units, and repetitively randomizing the distribution list during the distribution of the work requests to the distributed processing units as taught by Kapoor because this would allow to efficiently utilize the multiple web servers of an Internet host as well as reduce the skewed locking problems such that overall Internet traffic and response times are reduced (col. 2, lines 47-52).

32. **Claims 12-15 and 30-33 are rejected under 35 U.S.C. 103(a) as being unpatentable Komai (U.S. 2003/0187711 A1) (hereinafter Komai) in view of Greuel et al. (U.S. 7,003,564 B2) (herein after Greuel), further in view of Garnett et al. (U.S. 7,032,037) (herein after Garnett).**

33. As to claims 12 and 30, Komai teaches in a data processing network including a network file server and a plurality of virus checking server, a method comprising:

the network file server (server 1) obtaining a respective utilization value of each virus checking server (each personal computer, paragraph 58) (Figs. 2-3, display the schedule of the user uses the personal computer 2; for example, the user is scheduled to prepare a document at the user's seat from 10:00 to 12:00 on 17th (Thursday). Moreover, it is known that the same user is scheduled to receive the certifying examination held at the conference hall for a period from 13:00 to 15:00 on 17th

(Thursday), paragraph 79. Thus, Komai teaches displaying (obtaining) a respective utilization value of each virus checking server);

the network file server (server 1, paragraph 72) applying a function to the respective utilization value of said each server to obtain a respective weight for said each virus checking server (according to the schedule in Fig. 3, this user is scheduled to receive a certifying examination at a conference hall for 120 minutes from 13:00 to 15:00, immediately, the server 1 refers to an unoccupied time of 120 minutes, paragraphs 71-72);

34. Komai does not explicitly teach obtaining a respective weight by using a mapping function. However, Greuel teaches obtaining a respective weight by using a mapping function (for each system variable, a mapping by which a raw data value associated with the corresponding system variable is mapped to a score, and for each system variable, a weight, col. 2, lines 39-42 / a score mapped from the CPU utilization, col. 5, lines 4-29).

35. It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teaching of Komai by incorporating the teaching of a respective weight by using a mapping function as taught by Greuel in order to gain the advantage of monitoring of the health or status of a computer network to determine whether the computer network is online or offline (i.e., up or down) as a result of providing reliable, available and high performing computer networks and applications.

36. Komai and Greuel do not explicitly teaches the network file server using the respective weights for the virus checking servers for weighted round-robin load balancing of virus checking requests from the network file server to the virus checking servers. However, Garnett teaches using the respective weights for the servers for weighted round-robin load balancing of requests from the network file server to the servers (load balancing algorithm is the "weighted round robin" where the number of connections assigned per server is specified by a weight assigned to each server, col. 32, lines 13-16).

37. It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teaching of Komai and Greuel by incorporating the teaching of the respective weights are used for weighted round-robin load balancing of the work requests upon the distributed processing units as taught by Garnett in order to gain the advantage of preventing a particular application on the server is overloaded as sugguest by Garnett.

38. As to claims 13 and 31, Greuel teaches said each virus checking server collects statistics for calculation of the respective utilization value of said each virus checking server (the data collector collects a raw data value corresponding to the system variable, col. 2, lines 22-23; and col. 3, lines 38-40).

39. As to claims 14, 32, Greuel teaches the respective weight for said each virus checking server is programmed into a mapping table (table 233, Fig. 2B; col. 5, lines 4-29), and the network file server indexes the mapping table with said each respective utilization value to obtain the respective weight for said each virus checking server (the table 233 contains columns for the router name (or the address) (index), and the overall score in this example is computed as the weighted average of two numbers: (1) the interface health and (2) a score mapped from the CPU utilization, col. 5, lines 4-10).

40. As to claims 15 and 33, Garnett teaches the weighted round-robin load balancing performs round-robin load balancing when the weights are equal (col. 32, lines 16-18).

41. **Claims 16, 18, 34 and 36 are rejected under 35 U.S.C. 103(a) as being unpatentable Komai (U.S. 2003/0187711 A1) (hereinafter Komai) in view of Greuel et al. (U.S. 7,003,564 B2) (herein after Greuel), further in view of Garnett et al. (U.S. 7,032,037) (herein after Garnett) as applied to claims 12 and 30 above, and further in view of Kapoor (U.S. 5,884,038).**

42. As to claims 16 and 34, Komai and Greuel do not explicitly teach the respective weights for the virus checking servers are used for weighted round-robin load balancing of virus checking requests from the network file server to the virus checking servers by creating a distribution list containing entries indicating the virus checking servers. However, Garnett teaches the respective weights for the virus checking servers are

used for weighted round-robin load balancing of virus checking requests from the network file server to the virus checking servers by creating a distribution list containing entries indicating the virus checking servers (first server … next server in the list of available servers, col. 32, lines 4-25).

43. It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teaching of Komai, Greuel, and Garnett by incorporating the teaching of the respective weights for the virus checking servers are used for weighted round-robin load balancing of virus checking requests from the network file server to the virus checking servers by creating a distribution list containing entries indicating the virus checking servers as taught by Garnett in order to gain the advantage of preventing a particular application on the server is overloaded as sugguest by Garnett.

44. Komai, Greuel, and Garnett do not explicitly, the respective weight for said each distributed processing unit specifying the number of the entries indicating said each distributed processing unit, and by randomizing the distribution list, and accessing the randomized distribution list for distributing the work requests to the distributed processing units as indicated by the entries in the randomized distribution list.

45. However, Kapoor teaches the respective weight for said each distributed processing unit specifying the number of the entries indicating said each distributed

processing unit (the domain name server returns the IP address of a web server such that the total number of times that the IP address of each one of the web servers is returned in proportional to the relative weight of each web server, col. 5, lines 11-16), and by randomizing the distribution list (the domain name server randomize the list of web server, col. 5, lines 15-16 / the order of all the elements and array A are randomized, col. 5, lines 60-63), and accessing the randomized distribution list for distributing the work requests to the distributed processing units as indicated by the entries in the randomized distribution list (the domain name server receives a resolution request. The domain name server returns the IP address of a web server such that the total number of times that the IP address of each one of the web servers is returned in proportional to the relative weight of each web server. In order to return the IP address of a web server to client, the domain name server has to access to web server lists. Thus, Kapoor teaches accessing the randomized distribution list for distributing the work requests to the distributed processing units as indicated by the entries in the randomized distribution list).

46. It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teaching of Komai, Greuel, and Garnett by incorporating the teaching of the respective weight for said each distributed processing unit specifying the number of the entries indicating said each distributed processing unit, and by randomizing the distribution list, and accessing the randomized distribution list for distributing the work requests to the distributed processing units as indicated by the

entries in the randomized distribution list as taught by Kapoor in order to gain the advantage of efficiently utilize the multiple web servers of an Internet host as well as reduce the skewed locking problems such that overall Internet traffic and response times are reduced (col. 2, lines 47-52).

47. As to claims 18 and 36, Kapoor teaches the network file server obtains the utilization values of the virus checking servers at the start of a heartbeat interval (domain name server periodically determines whether each one of the web server is not functioning proper or is generating time out errors by reading a list of IP addresses of web servers for a given host and their relative weight, col. 5, lines 23-42) , randomizes the distribution list repetitively during use of the distribution list for load balancing of virus checking requests during the heartbeat interval (array A is periodically recomputed as desired, col. 5, lines 63-64), obtains new utilization values of the virus checking servers at the start of a following heartbeat interval (domain name server periodically determines whether each one of the web server is not functioning proper or is generating time out errors by reading a list of IP addresses of web servers for a given host and their relative weight, col. 5, lines 23-42), and produces a new distribution list from the new utilization values of the virus checking servers for load balancing of virus checking requests during the following heartbeat interval (domain name server connects to each web server and removes the down web servers from the list , col. 5, lines 42-44).

48. **Claims 17 and 35 are rejected under 35 U.S.C. 103(a) as being unpatentable Komai (U.S. 2003/0187711 A1) (hereinafter Komai) in view of Greuel et al. (U.S. 7,003,564 B2) (herein after Greuel), further in view of Garnett et al. (U.S. 7,032,037) (herein after Garnett) as applied to claims 16 and 34 above, further in view of Kapoor (U.S. 5,884,038), and further in view of Grochowski (U.S. 6,115,807).**

49. As to claims 17, 35, Komai, Greuel, Garnett , and Kapoor do not explicitly teach re-randomizing the distribution list for re-use once the end of the distribution list is reached during the distributing of the work requests to the virus checking servers as indicated by the entries in the randomized distribution list. However, Grochowski teaches re-randomizing the distribution list for re-use once the end of the distribution list is reached during the distributing of the work requests to the virus checking servers as indicated by the entries in the randomized distribution list (if the decoder comes to the end of the queue, it rotates around to the beginning of queue (col.2, lines 65-67). As to the limitation of "randomize", Grochowski teaches "rotating". Noting that randomizer is also defined as rotator. Extrinsic evidence, inventor name (U.S. 6,819,276), disclose randomizer is also defined as rotator (col. 5, lines 46-47).

50. It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teaching of Komai, Greuel, Garnett, kapoor by incorporating the teaching of re-randomizing the distribution list for re-use once the end of the distribution list is reached during the distributing of the work requests to the virus

checking servers as indicated by the entries in the randomized distribution list as taught by Grochowski in order to gain the advantage of determining how many of the decoded instructions may issue in a next clock cycle thereby to ensure that neither too many nor too few instructions enter pipeline at any given time thereby improve the effective of processors.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to CAMQUY TRUONG whose telephone number is (571)272-3773. The examiner can normally be reached on 9:00am - 5:30pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng Ai An can be reached on (703)305-9678. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Meng-Ai An/
Supervisory Patent Examiner, Art Unit 2195

Camquy Truong

August 22, 2008